

# Hands-On Simulations of Interfacial Electron Transfer

Victor S. Batista

*Department of Chemistry, Yale University, P.O. Box 208107, New Haven, CT 06520-8107*

This document with embedded links aims to illustrate an introductory 'hands-on' tutorial module on computational tools that provide atomic-level understanding of electron transfer induced by photoexcitation of molecules on surfaces, as in dye-sensitized solar cells (DSSCs), electrical nanodevices for bio-detection and receptors, or interfacial chemistry of catalysis. We focus on interfacial electron transfer (IET) in sensitized  $\text{TiO}_2$  surfaces, allowing the students to gain experience in quantum dynamics based on the Extended-Hückel method by using the software package IETsim, developed in the Batista research group as well as other packages such as Gaussian, VMD and Yaehmop. Avogadro could also be used for all visualization purposes of this tutorial.

The tutorial illustrates several innovative aspects for undergraduate education. First, the exercise introduces DSSCs, a current hot topic in physical chemistry and technology that to the best of our knowledge is not mentioned by any current textbook. It applies first principles methods to understand the properties of sensitized semiconductor materials, relevant to the development of new technologies for the environment and energy, and pushes the discussion of several concepts of modern physical chemistry (electron transfer, quantum mechanics, etc.) to the broader area of science and technology. Furthermore, the exercise aims to familiarize the student with quantum dynamics which is a topic not addressed by current textbooks, using new theoretical and computational tools that provide atomic-level understanding into processes relevant for applications such as solar devices, new semiconducting topologies, electrical nanodevices for biodetection and receptors, interfacial chemistry of catalysis, and electron transfer. The homework also provides an opportunity to actually perform operations in quantum mechanics and coordinate the exercise with the content of the lectures as well as with more extended presentations, illustrated with the use of links to movie, and supplementary materials with up-date-references.

IET involves the injection of an electron from a molecular adsorbate into an electrode surface, as occurs when photoexcited electrons in an adsorbate molecule are transferred into the conduction band of  $\text{TiO}_2$  in the photoanodes of dye-sensitized solar cells (see Fig. 1).

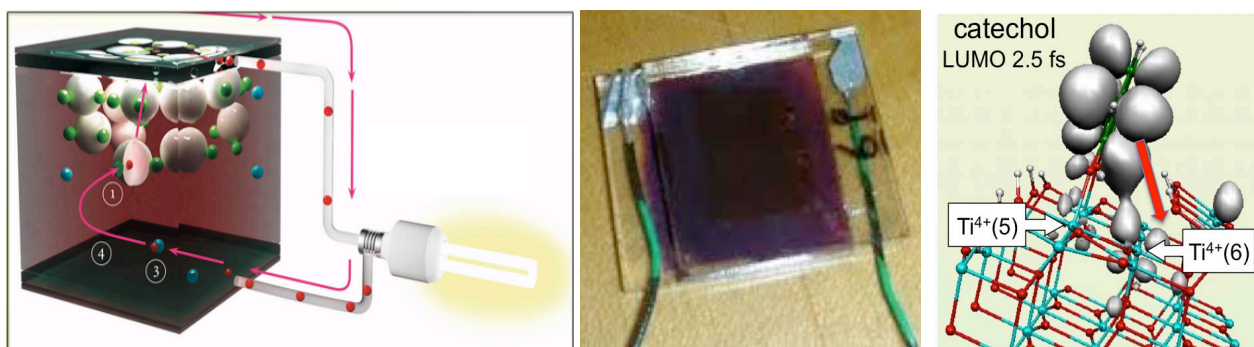


Figure 1: Left panel: Diagram of a DSSC. Middle panel: DSSC based on  $\text{TiO}_2$  sensitized with red-dye (see movie). Right panel: Atomistic description of IET from a catechol adsorbate (green) into the conduction band of  $\text{TiO}_2$  (Ti: blue; O:red; H: white;  $e^-$  density: gray).

A detailed description of the computational methodology applied for simulations of IET can be found in references [1], [2], [3], and [4]. Here, we outline the methodology as implemented for a fixed nuclear configuration in the IETsim package, and we illustrate it with the specific example shown in Fig. 1 (*i.e.*, IET from a catechol adsorbate into  $\text{TiO}_2$ ). The tutorial aims to encourage students to continue working in the field, or at least to have something of an insider's point of view as the field develops in the next few decades.

IETsim expands the initial electronic state  $|\Psi(0)\rangle$  (e.g., the photoexcited electron in the adsorbate molecule) as a linear combination of atomic orbitals  $|K_j\rangle$  and evolves it in time by integrating the time dependent Schrödinger equation analytically, as follows:  $|\Psi(t)\rangle = \sum_k \langle \phi_k | \Psi(0) \rangle e^{-\frac{i}{\hbar} E_k t} |\phi_k\rangle$ . Here,  $|\phi_k\rangle = \sum_j C_{j,k} |K_j\rangle$  is the eigenstate with eigenvalue  $E_k$  obtained by solving the extended-Hückel (EH) eigenvalue problem  $\mathbf{HC} = \mathbf{SCE}$  in the basis of atomic orbitals  $|K_j\rangle$  with  $S_{ij} = \langle K_i | K_j \rangle$ . The EH problem is solved by using the program Yaehmop developed by the research group of Roald Hoffmann at Cornell University. The EH Hamiltonian is an improvement over the familiar Hückel method taught in undergraduate P-chem courses (e.g., Theories of Electronic Molecular Structure). In the Hückel method, only  $\pi$  orbitals are included in the description of valence electrons, using the basis set of P atomic orbitals to give a qualitative description of conjugated hydrocarbons. The EH hamiltonian expands the Hückel basis set and includes both  $\sigma$  and  $\pi$  orbitals formed by S, P and D atomic orbitals, representing the full valence shell of a wide range of molecular structures. The EH method is a good starting point for simulating IET in sensitized semiconductor surfaces because it is computationally efficient and provides a semi-quantitative description of IET processes central to photoconversion in DSSC's.

The tutorial is organized as follows: Section 2 details the installation of IETsim on a Linux system. Section 3 describes the input file. Section 4 shows how to run an example calculation. Section 5 maps the construction of a few key figures, similar to those presented in Reference [1], using the output file from IETsim and highlights the ability of the methodology to provide insight on IET at the molecular level. Finally, Section 6 describes a general procedure for generating new input files and sensitizer-TiO<sub>2</sub> nanostructures.

## 1 The Exercise

Consider a TiO<sub>2</sub> slab with atomic coordinates define in file Tio2.com. Download the software package IETsim and compute:

1. The distribution of electronic states as a function of energy, as described by density of states (DOS) of TiO<sub>2</sub>.
2. The DOS of TiO<sub>2</sub> sensitized with catechol covalently attached to the (101) surface, as shown in Fig. 3 (also, Figure 6 in Reference [1]).
3. The dynamics of electron transfer as described by the time-dependent electronic population of catechol PMOL(t), when the initial state is defined as the LUMO+1 orbital of the isolated catechol on the TiO<sub>2</sub>-anatase (101) surface. Plot the survival amplitude and estimate the ET rate. Compare your result with Figure 4 (also, Figure 13 in Reference [1]).
4. Simulate the IET from the LUMO orbital of catechol on the TiO<sub>2</sub>-anatase (101) surface. Explain why the probability PMOL(t) does not decay to zero. Compare your results to the following picture (Fig. 2):

### Solution to the Exercise:

Follow the instructions in the tutorial below to install, compile and run IETsim using the input file provided by the embedded links. The tutorial also provides guidelines to construct figures of the *DOS*, the time evolution of the electronic density during IET and the time-dependent electronic population.

## 2 Compiling IETsim

The source code of IETsim is available as a zipped tar file, dynamics-0.4.tar.gz. A tar file is used to enclose a filesystem tree, preserving the files and their enclosing directories. To open the tar file, type:

```
$ tar -xzf dynamics-0.4.tar.gz
```

where the switches used are x for extract, z for compressed file and f indicating the filename will follow. The directory dynamics-0.4 will be created and you should move into the directory using the command

```
$ cd dynamics-0.4
```

The following directories will be present: cubebuilder, docs, dynamics, examples and kpoints. Remember, you can use the command

```
$ ls
```

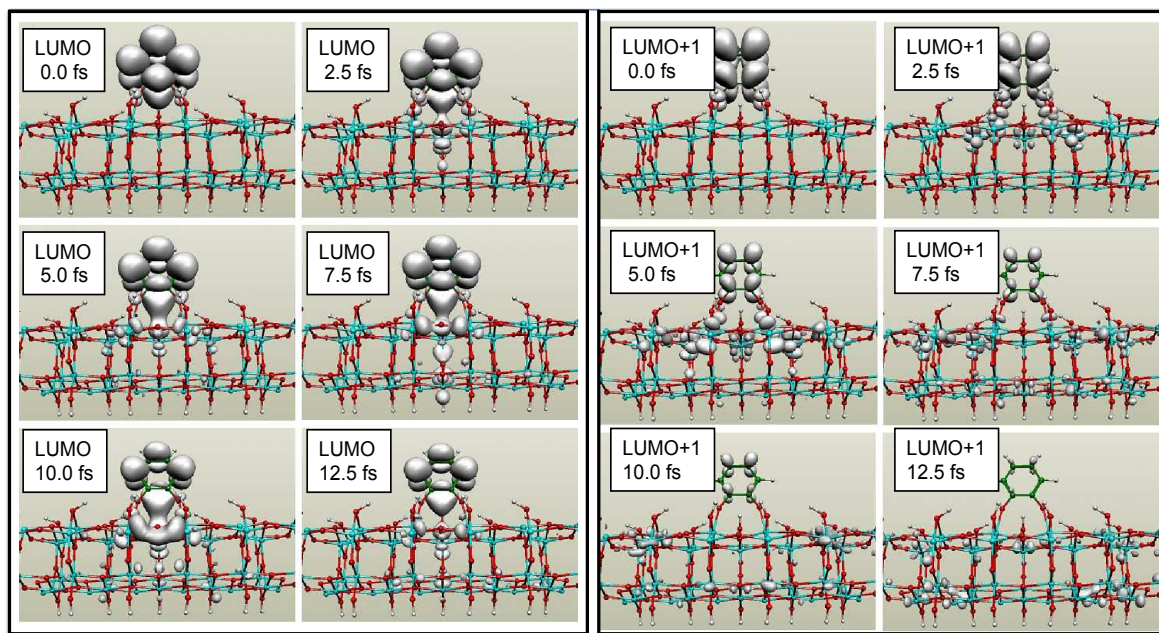


Figure 2: Atomistic description of IET from a catechol adsorbate HOMO (left) and LUMO (right) into the conduction band of  $\text{TiO}_2$  (Ti: blue; O:red; H: white;  $e^-$  density: gray).

to view a list of files and directories in the current working directory. There are a few codes in the different directories that will need to be compiled before you can complete the tutorial.

We begin with compiling IETsim that includes Yahemop and the module to simulate IET. You need to move into the directory dynamics, by typing:

```
$ cd dynamics
```

and you should be able to find the file, Makefile. The Makefile is used to compile the code but before you can do that you will need to change one line in the makefile. The 11<sup>th</sup> line of the code currently reads

```
PARM_FILE_LOC = \"/home/robert/dynamics-0.4/dynamics/eht_params.dat\"
```

since it is assumed the tar file was downloaded in directory /home/robert/. You need to change '/home/robert' by the name of the directory where you downloaded the tar file in your filesystem. To find the name of your current working directory, use the command

```
$ pwd
```

which in the current case outputs

```
/home/robert/dynamics-0.4/dynamics
```

but in your case would be different. Copy the analogous output from applying that command in your system and paste in place of /home/robert/dynamics-0.4/dynamics in your Makefile. This can be done by using a convenient text editor (*e.g.*, nedit, emacs, vi, etc.). After editing the Makefile, compile the code by using the command

```
$ make
```

which would be followed by copious amounts of output printed to the terminal. Warnings will appear in the output and they can be ignored, but if you see a message indicating an error then the code did not compile properly. You should consult your system administrator for help with compiling the code.

The next code that needs to be compiled is used to extract the density of states (DOS) information from the Yahemop output file. The name of the code is "fitdos" that will be discussed in the following sections. You can compile the code by changing directory to /dynamics-0.4/dynamics/utlis,

```
$ cd utlis
```

and using the Makefile to compile

```
$ make
```

generating a few warnings that can be ignored.

The final code that needs to be compiled is called cubebuilder. This code is used to generate cube files that allow the electron density isosurface to be visualized with VMD or GaussView. You should change directory into dynamics-0.4/cubebuilder

```
$ cd ../../cubebuilder
```

and once again use the Makefile to compile the code

```
$ make all
```

which is a much smaller code and should have minimal output. If you were confused by the previous change directory command, that is because

```
$ cd ../
```

moves to the parent directory of the current working directory. Therefore,

```
$ cd ../../
```

moves up two directories in the directory tree from your current working directory.

### 3 Input file

This section describes the input file of IETsim with emphasis on how to setup calculations of the electronic density of states (DOS) and simulations of IET. The input file typically has the file extension .bind, but file extensions are not as important as in Windows. Inside the .bind file is the molecular structure, unit cell dimensions, kpoint sampling scheme, etc. An example of an input file is available in the directory dynamics-0.4/examples. The file can be viewed using 'less', a linux tool used to view files. To view the file with 'less', use the command:

```
$ less catechol_anatase101.bind
```

when your current working directory is the 'example' directory. To quit the 'less' text-viewer, simply type the letter 'q'. If you need to modify the file do it with a text editor (*e.g.*, nedit, emacs, vi, etc.) since unfortunately 'less' is a read-only command.

#### 3.1 Molecular Structure

The beginning part of the file 'dynamics-0.4/dynamics/examples/catechol\_anatase101.bind' gives the description of the molecular structure coordinates. The structure can either be formatted in cartesian coordinates, which is convenient for a molecular structure, or in crystallographic coordinates for periodic structures. In this tutorial we will focus on periodic slabs of TiO<sub>2</sub> so we will use crystallographic coordinates. Crystallographic coordinates give the location of atoms in the periodic cell in fractions of the crystal vectors. In Section 6, you will be guided through an example for generating crystallographic coordinates from cartesian coordinates. Lets take a look at the first 10 lines of the file catechol\_anatase101.bind and discuss the function of each line,

```
catechol_anatase101.com
```

```
Geometry Crystallographic
```

```
127
```

1	O	0.467536	0.404248	0.202821
2	O	0.467125	0.596532	0.202983
3	C	0.505225	0.453304	0.156686
4	C	0.505082	0.547690	0.156798
5	C	0.545836	0.410086	0.104319
6	C	0.545685	0.591452	0.104593

The first line is a comment line so you can write anything you want to keep track of the action of the input file. For example, you can enter the name of the .com file generated by GaussView, when you use the graphics interface GaussView to generate the atomic coordinates of the molecular system, as discussed later in this tutorial. The second line is a blank line and, yes, it is required. The third line indicates that the geometry will be formatted in crystallographic coordinates. In this case, there are 127 atoms in the file. Actually, there are only 124 because the last 3 are the so-called 'dummy atoms' that specify the crystal unit vectors, as discussed later in this section. The fourth line gives the total number of atoms in the system plus, including dummy atoms. The next 6 lines describe the first six atoms of the geometry. In this case, they correspond to the oxygens and four of the carbons of catechol. All atomic entries need to follow the same format: atom number, atom symbol and the xyz coordinates.

Now we discuss the dummy atoms at the end of the list of coordinates. The dummy atoms are used to specify the periodic lattice. Their use will be discussed in Section 3.3. Dummy atoms are defined by the atomic symbol '&', as seen at the end of the input file:

```
123  H    0.704176    0.499003    0.557056
124  H    0.702664    0.748596    0.557086
125  &    1.467536    0.404248    0.202821
126  &    0.467536    1.404248    0.202821
127  &    0.467536    0.404248    1.202821
```

where atoms 125, 126 and 127 are defined as dummy atoms with atomic coordinates equal to the coordinates of atom 1, incremented by 1.0 in the x, y and z coordinates, respectively.

### 3.2 Molecular Charge Keyword

The molecular charge specifies the overall electrostatic charge of the system. The format to specify the charge to be zero is given in lines 133 – 134 of the file 'dynamics-0.4/dynamics/examples/catechol\_anatase101.bind':

```
Charge
0
```

The keyword 'Charge' is followed by a new line with the value of the charge (e.g., 0 in this case).

### 3.3 Periodic Lattice Keyword

When you have a periodic system, you need to specify the lattice vectors. The way this is accomplished in Yaehmop is by using atoms in the geometry. In the example input, file the 'Lattice' keyword and associated data can be found on lines 136 – 147. Here are those file lines

```
Lattice
3
1 1 1
1 125
1 126
1 127

Crystal Spec
10.227840 15.128000 22.197750
90 90 90
```

The first line gives the keyword. The second line gives the number of lattice vectors. In this case there is 3 because the system is periodic in 3 dimension. Actually, the system is periodic in only 2 dimension (2D), but a 2D system can be modeled as a 3D system by making the magnitude of the lattice vector in the non-periodic direction larger than the length of the molecular structure. The third line indicates the number of cell images used when calculating the overlap matrix for the atomic orbital basis. As long as you use a value  $\geq 1$  you should be fine, but realize that the calculation time will increase with increasing cell images. The next three lines are where those dummy atoms we talked about in the Section 3.1 enter into play. The three lines indicate that the three lattice vectors are the vectors defined as the vector from atom 1 to 125, 1 to 126 and 1 to 127.

The position of the dummy atoms were constructed by adding 1.0 to either the x, y or z coordinate of atom 1. Therefore, the first lattice vector corresponds to the x axis, the second the y axis and finally the third is the z axis.

The second keyword in this section is 'Crystal Spec' which is used to indicate the length of the lattice vectors on the following line. You can see that the length of the third lattice vector is 22.197750, indicating the length of the periodic cell in the z direction. The final line gives the angle between the lattice vectors. A value of 90 degrees between all the unit vectors indicates a rectangular periodic cell.

### 3.4 IET Dynamics Keyword

Finally, we come to the first keyword used in simulating IET. The "Dynamics" keyword gives the timestep and the length of the simulation. The keyword and needed information is

```
Dynamics
0.1 100.0
```

in the example input file and can be found on lines 149 – 150. The first number on the second line is the timestep. The timestep of 0.1 is in units of fs. The second number on the second line is the total simulation time desired. In the example input file, the requested simulation time is 100.0 fs.

### 3.5 Adsorbate Keyword

The next keyword used for IET simulations is the 'Adsorbate' keyword. This is where the atom numbers of the adsorbate and the initial adsorbate orbital is specified. The keyword and corresponding information should be formatted as

```
Adsorbate
12 22 1.0
1,2-12
```

which can be found in the example input file on lines 153 – 155. The second line gives three numbers which are the number of atoms in the adsorbate (catechol has 12 atoms), the adsorbate orbital to initialize the photoexcited electron (orbital 22 corresponds to the LUMO for catechol), and a weighting factor for the initial orbital. A contribution from a second adsorbate orbital can be introduced by replacing the second line with the following

```
12 22 0.5 23 0.5
```

which gives equal weight to both orbitals and the second orbital introduced is the catechol LUMO+1. Catechol has 12 atoms when both oxygens are deprotonated. When catechol binds to the TiO<sub>2</sub> surface, the hydrogens on the oxygens are replaced with titaniums from the surface. The LUMO orbital is orbital number 22 because catechol with an overall molecular charge of -2 has 42 electrons. Remember, there are two electrons per orbital! The third lines gives the atom number of catechol in the molecular structure. Give the atom numbers separated by commas, or if they are consecutive, use a hyphen as in the example. If you do not know the atom number of the adsorbate, use GaussView to visualize the adsorbate and record the atom numbers.

### 3.6 Occupation Keyword

The "Occupation" keyword asks the IET module to monitor the probability that the electron is located on a collection of atoms during the evolution. The example given in the example inputfile is

```
Occupation
112
13,14-124
```

which can be found on lines 157 – 159. This keyword is followed by the number of atoms on the second line. In the example there are 112 atoms, corresponding to all the atoms except the adsorbate. The third line gives the atom numbers. This keyword is useful if you want to look at the population of the photoexcited electron on particular moieties of the adsorbate during IET.

### 3.7 Cube Keyword

The 'Cube' keyword tells the IET module that you want to generate cube files of the electron density evolution. The keyword is just "Cube" without further input and can be found in the input file as

```
Cube
```

The auxillary code cubegen is used to generate the cubefiles but information is required from Yaehmop. The way to information is saved is by the creation of a .wave file when the "Cube" keyword is present. If you do not want to generate the .wave file (you don't want to generate cube files), then you can remove the 'Cube' keyword. Not generating the .wave file should help speed-up the code since writing a rather large file is avoided. Running the cubegen code will be discussed in Section 5.3.

### 3.8 Density of States Keyword

A DOS plot is very helpful for identifying where the energetics of the adsorbate excited states are poised compared to the TiO<sub>2</sub> conduction band. DOS information is included in the output file when the following is present in your input file

```
Average Properties

Projected DOS
1
atoms 1 1,2 1,3 1,4 1,5 1,6 1,7 1,8 1,9 1,10 1,11 1,12 1
```

as can be found in the example input file one lines 163 – 167. The first line tells the code to calculate average properties. When average properties are requested, you need to include the 'Kpoints' keyword discussed in the next section. If you want to perform a projected DOS calculation, you need to include the appropriate keyword found on the third line above (line 165 of the example input file). The 'projected DOS' keyword is followed by the number of projected DOS's, in the example only one is requested. The fifth line (line 167 of the example input file) indicates which atoms will be used in the projected dos. The line must start with the word 'atom' followed by a list of comma separated numbers. There are two number between each comma with the first indicating the atom number and the second indicating the weight due to that atom. You should typically use a weight of 1.0 corresponding to that found in the example input file.

The DOS results available in the output file can be extracted in a form ready to be plotted using the fitdos code. The code will be discussed further in the Output Files and View Results section.

### 3.9 Kpoints Keyword

The 'Kpoints' keyword is required for average properties calculations and IET simulations of periodic systems. The kpoints are points in the reciprocal lattice used to calculate average properties of the system. In this tutorial, we will restrict ourselves to a single kpoint at the origin with a weight of 1.0. The way to specify a single kpoint at the origin with a weight of 1.0 is as follows

```
kpoints
1
0.000000 0.000000 0.000000 1
```

which can be found in the input file. A larger number of kpoints gives better results for DOS calculations and other properties but the generation of the kpoints is beyond the scope of this tutorial. Therefore, the 'Kpoints' keyword should always be given in the shown format and the format will not be discussed in more detail. Those interested in using more kpoints should consult the Yaehmop user manual called `bind_manual.pdf`, also found in the directory `'/dynamics/dynamics-0.4/docs'`.

### 3.10 Absorbing Boundary Keyword

To model the surface of TiO<sub>2</sub>, we have used a small periodic slab in the example input file. Due to the finite size of the slab, the photoexcited electron will artificially return to the adsorbate. The artificial recurrences can be reduced by making the slab larger. Unfortunately, the calculation becomes too slow when more atoms are

included in the molecular structure. A way to circumvent this problem is to include 'absorbing potentials' that absorb (i.e., damp) the electron density when the scattering electron populates the orbitals of the atoms at the base of the TiO<sub>2</sub> slab. Such an absorbing potential mimics the effect of the rest of the TiO<sub>2</sub> surface, absorbing the electronic density out of the simulations box. The 'Absorbing' keyword requires a format specified in lines 175 – 184 of the input file: Looking at those lines

```
Absorbing
72 1.000
118, 119, 120, 121, 122, 123, 124, 125, 126, 136 \
137, 138, 139, 140, 141, 142, 143, 144, 154, 155 \
156, 157, 158, 159, 160, 161, 162, 172, 173, 174 \
175, 176, 177, 178, 179, 180, 253, 254, 255, 256 \
257, 258, 259, 260, 261, 262, 263, 264, 265, 266 \
267, 268, 269, 270, 271, 272, 273, 274, 275, 276 \
277, 278, 279, 280, 281, 282, 283, 284, 285, 286 \
287, 288
```

we see there are many numbers. The first line after the 'Absorbing' keyword has two numbers. The first number is the number of orbitals with the absorbing potential and the second number is the value of the absorbing potential. The absorbing potential creates an exponential damping of the electron population. The larger the damping factor, the faster the electron population will be damped in the orbital. The timestep is included in the exponential damping so a smaller timestep with the 'Dynamics' keyword will result in quicker damping. The next eight lines list the orbital numbers where the damping will be applied. You should have the same number of orbitals as stated on the previous line. In this case there are 72 orbitals. Remember that a Ti atom has 9 valence orbitals, so 72 orbitals located on Ti atoms correspond to 8 atoms. The eight Ti atoms are atom numbers 22, 24, 26, 28, 37, 38, 39 and 40 in the molecular structure. You can check that these Ti atoms correspond to those in the bottom of the slab, by opening the .com file available in the examples directory with GaussView. The orbital numbers need to be separated by commas. The backslashes indicate that more orbital numbers will follow on the next line.

To ease the construction of the 'Absorbing' keyword input, a python script is available in the dynamics directory. The python script requires as input the .com file and a list of atoms where you want absorbing potentials. From the examples directory, the command to generate the input found in the example input file is

```
$ cd ~/dynamics-0.4/examples
$ ../dynamics/prep_absorbing.py catechol_anatase101.com 22 24 26 28 37 38 39 40
```

and you can copy and paste the output into the input file that you are generating.

Gaussview or Avogadro can also visualize labels.

### 3.11 More Information

More information about the Yaehmop input file can be found in the user manual located in the docs directory. You should be able to open the manual using xpdf

```
$ cd ~/dynamics-0.4/docs
$ xpdf bind_manual.pdf
```

The user manual is for the unmodified version of Yaehmop. Therefore, no information regarding the simulation of IET is enclosed.

## 4 Running the Example

In this section we will discuss how to run the code so we can start generating results. The first thing that I suggest you do is create a new directory to keep the generated files to help your file system stay neat and organized. You should change directory back to your home directory and then create a new directory called 'tutorial'

```
$ cd ~/
$ mkdir tutorial
$ cd tutorial
```



From here, you should copy the bind input file available in the examples directory

```
$ cp ~/dynamics-0.4/examples/catechol_anatase101.bind .
```

and the period is important at the end of the command because it tells the copy command to copy the file into the current working directory. Now, launch the executable with the input file

```
$ ~/dynamics-0.4/dynamics/dynamics catechol_anatase101.bind
```

and output should be written to the terminal. The code will finish with

Done!

indicating that the execution ended properly. The code should run for about a minute before it finishes. Finally, it is time to analyze the results.

## 5 Output Files and Visualizing Results

After running Yaehmop with the input file, catechol\_anatase101.bind supplied in the examples directory, you will notice that a couple new files are present in your working directory. In the following, we will discuss how you can use these output files to visualize results.

### 5.1 Generating a Density of States Plot

To generate a DOS plot we will use the .out file produced by Yaehmop and the fitdos tool. To run the fitdos tool use the command

```
$ ~/dynamics-0.4/dynamics/utils/fitdos catechol_anatase101.bind
```

from the Tutorial directory. The tool will ask you to enter: E min, E max, broadening and Energy Step. The values I suggest you use are

```
Enter E min: -18.0
Enter E max: -6.0
Enter broadening: 50
Enter Energy Step: .01
```

The min and max set the energy range for the DOS. The energy step gives the grid size for the plot of the DOS. The broadening specifies the width of gaussians used to convolute the plot. The convolution is used to produce a smooth DOS plot.

The fitdos tool will produce the file catechol\_anatase101.bind.DOS. The number of columns in the .DOS file depends on the number of projected DOS's requested. In the example, 1 projected DOS was requested so there are 3 columns. The first column is the DOS, the second is the projected DOS, and the final column is the energy range. You should plot the DOS using your preferred plotting package.

I will show you how to make the plot using gnuplot. The following commands will generate Figure 3

```
$ gnuplot
gnuplot> set xrange [-18:-6]
gnuplot> set xlabel "ENERGY (eV) "
gnuplot> set ylabel "DOS"
gnuplot> set format y ""
gnuplot> set style line 1 lt -1 lw 0.8
gnuplot> set style line 2 lt -1 lw 2
gnuplot> plot "catechol_anatase101.bind.DOS" u 3:1 w l ls 1 t "DOS"
gnuplot> replot "catechol_anatase101.bind.DOS" u 3:2 w l ls 2 t "pDOS"
```

When you are ready to leave gnuplot, type the command exit

```
gnuplot> exit
$
```

You can compare your results to Figure 3 and with the corresponding Figure 6 in Reference [1].

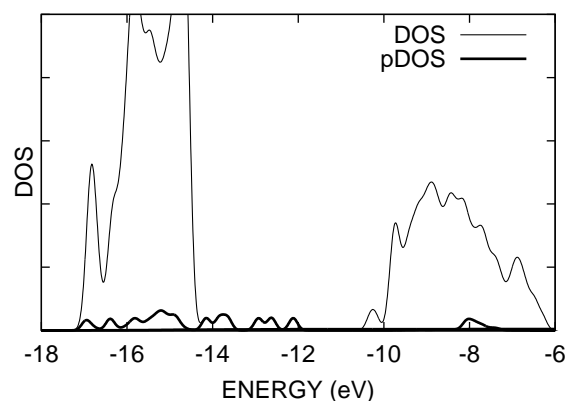


Figure 3: Density of states of the catechol on  $\text{TiO}_2$  nanostructure. The thick curve at the bottom of the bottom of the graph represents the projected density of states on the catechol basis functions. The thin curve is the total density of states for the full nanostructure.

## 5.2 Visualizing the IET Curves

The results from the IET simulation are output into a `.bind.edyn` file. For the current example, the filename is `catechol_anatase101.bind.edyn`. In the file there are four columns. The first column is the simulation time. The second column is the survival probability. The survival probability gives the density of the photoexcited electron remaining on the adsorbate. The third column is the norm of the wavefunction for the photoexcited electron. When absorbing potentials are not in use, the norm should remain at 1.0. However, when absorbing potentials are in play, the norm of the wavefunction will decrease. The fourth and final column is the occupation (or density) of the photoexcited electron on the atoms you selected with the 'Occupation' keyword (Section 3.6).

Using the `.bind.edyn` file produced with the example input file for catechol on  $\text{TiO}_2$ , we can plot the survival probability as a function of time using gnuplot with the following commands

```
$ gnuplot
gnuplot> set xrange [0:100]
gnuplot> set yrange [0:1]
gnuplot> set xlabel "TIME (fs)"
gnuplot> set ylabel "{/Symbol r}_{MOL} (t)"
gnuplot> set style line 1 lt -1 lw 0.8
gnuplot> set style line 2 lt -1 lw 0.8
gnuplot> plot "catechol_anatase101.bind.edyn" u 1:2 w l ls 2 t ''
gnuplot> replot exp(-x/6) w l ls 1 t "{/Symbol t} = 6 fs"
```

The resulting output should be the same as Figure 4. Remember, you can leave the gnuplot environment using the command "exit". You can compare Figure 4 with the corresponding Figure 8 in Reference [1].

## 5.3 Visualizing the Electron Density

To view the density of the photoexcited electron, we need to generate cube files and open the resulting files with either VMD or GaussView. The code we compiled earlier called `cubebuilder` is used to construct cube files from the `.wave` output file. The 'Cube' keyword needs to have been included in the input file otherwise the `.wave` file will not be generated. The `cubebuilder` needs its own input file. An example input file is available under the name `/dynamics-0.4/examples/catechol.in`.

Taking a look at the file `/dynamics-0.4/examples/catechol.in`, we see:

```
region
0.25
0.0000 10.22784
0.0000 15.12800
0.0000 22.19775
```

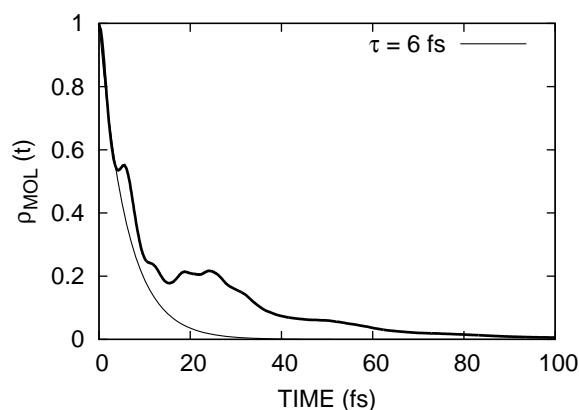


Figure 4: Time-dependent survival probability in a catechol sensitized TiO<sub>2</sub> nanostructure. The electronic excitation is initialized by populating the catechol-LUMO. The thin line is an exponential fitting to the elementary step.

```
make
10
1,2-10

name
cat_tio2
```

The parameters above specify 10 cubes with 1,2-10 to visualize 1-10 first 0.9 fs. The 'region' keyword specifies the size of the box. The electron density is output on a regular grid. The first number after the keyword is the distance between grid points. The second line is the minimum and maximum range on the x axis. The third line is the minimum and maximum for the y axis and the fourth line is for the z axis. The 'make' keyword determines how many cube files are generated. The first number after the keyword is the number of cube files. The next line gives the step number in the IET evolution with time zero,  $t = 0$ , corresponding to step 1. In this example we will make cube files of the first 0.9 fs. The 'name' keyword gives a basename for the cubefiles. The filename will be constructed using the basename plus the step number and finished with the .cube file extension, cat\_tio2\_1.cube for example.

Using the cubebuilder input file, the code can be run to generate the cube files. The .wave file produced from running Yaehmop with the example input file is called catechol\_anatase.bind.edyn.wave. Assuming you are in the Tutorial directory, run the commands

```
cp ~/dynamics-0.4/examples/catechol.in .
$ ~/dynamics-0.4/cubebuilder/cubebuilder catechol.in catechol_anatase.bind.edyn.wave
```

Generating cube files can take a significant amount of time, especially if you are producing many of them. You will need to be patient. The first cubefile generated from the example catechol on TiO<sub>2</sub> system, cat\_tio2\_1.cube, is the catechol-HOMO initial state used in the IET simulation. Figure 5 shows the photoexcited electron density. You can compare the electron density with Figure 11 in Reference [1].

The cube files can be opened with either VMD or GaussView. Here we will walk through the procedure for visualizing the cube file with GaussView. Open the cube file using the command

```
$ gview cat_tio2_1.cube
```

then follow these steps:

1. Click 'Results'
2. Select 'Surfaces'
3. In the 'Surfaces and Cubes' window Select the 'Surface Actions' button
4. Select 'New Surface'

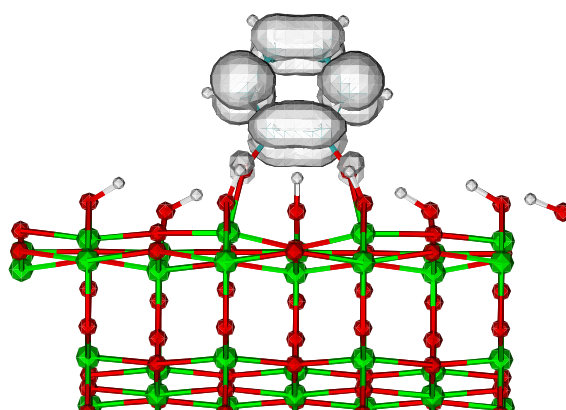


Figure 5: Charge distribution after populating the catechol LUMO.

The surface should appear in the molecule window.

To visualize the cube file in VMD, open the cube file using the command

```
$ vmd cat_tio2_1.cube
```

then follow these steps:

1. Click 'Graphics'
2. Select 'Representations'
3. In the 'Graphical Representations' window, Click 'Create Rep'
4. In the 'Graphical Representations' window, Change 'Drawing Method' from 'Lines' to 'Isosurface'
5. Change 'Draw' from 'Points' to 'Solid Surface'
6. Change 'Isovalue' from 0 to 0.002 by dragging the bar next to the value.

## 6 Generating Your Own Structure

The .com file used to create the example input file is available in the examples directory. You can open the file using GaussView

```
$ gview catechol_anatase101.com
```

and make modifications to the catechol adsorbate using the structure building features. You can also delete the catechol adsorbate and insert your favorite dye.

### 6.1 Preparing a Catechol TiO<sub>2</sub> Structure

To prepare a new catechol on TiO<sub>2</sub> structure we will first optimize the catechol adsorbate in vacuum. The resulting gas phase optimized structure will be adsorbed on the surface by hand using GaussView.

The catechol molecule can be constructed using GaussView. Open GaussView using the following command

```
$ gview
```

and follow these steps

1. Click 'View' then 'Builder'
2. Click 'File' then 'New', a new window 'G1:M1:V1 New' pops out for building structures.
3. Choose the 'Ring Fragment' icon (the 1st one in the 2nd row) in the 'Builder' window
4. Click on the inside the window of 'G1:M1:V1 - New', and then a benzene ring is build.
5. Select 'O' atom (by clicking the 1st icon in the 1st row) and click on the H-atom of benzene ring in 'G1:M1:V1 - New' to make a structure of catechol.
6. delete the two H atoms on the O atoms to make an anion (-2 charge) of catechol.
7. Click 'File' then 'Save', and save the structure as 'catechol.com'.

The file catechol.com should look like the following

```
%chk=Catechol.chk
# hf/3-21g geom=connectivity

Title Card Required

0 1
C          0.63636364    1.61157022    0.00000000
C          2.03152364    1.61157022    0.00000000
C          2.72906164    2.81932122    0.00000000
C          2.03140764    4.02783022   -0.00119900
C          0.63658264    4.02775222   -0.00167800
C         -0.06101836    2.81954622   -0.00068200
H          3.82874164    2.81940122    0.00063400
H          2.58160764    4.97997322   -0.00125800
H          0.08646064    4.98003322   -0.00263100
H         -1.16062236    2.81972922   -0.00086200
O         -0.07857634    0.37311938    0.00058521
O          2.74610797    0.37291518    0.00171004

1 2 1.5 6 1.5 11 1.0
2 3 1.5 12 1.0
3 4 1.5 7 1.0
4 5 1.5 8 1.0
5 6 1.5 9 1.0
6 10 1.0
7
8
9
10
11
12
```

Delete the first 6 lines of the file 'catechol.com' and replace with these 7 lines.

```
%mem=500mw
%nproc=2
# b3lyp/6-31g* opt=tight

geometry optimization: comment line

-2 1
```

You may do this using any editors of your choice, like 'emacs' or 'vi'. Submit the job for geometry optimization using Gaussian 09

```
$ g09 catechol.com
```

An output file, catechol.log, is generated once the geometry optimization is complete.

Build the structure where the catechol molecule (anion) is attached to the TiO<sub>2</sub> anatase surface with a bi-dentate binding mode. Figure 6 shows an image of the nanostructure that you are building.

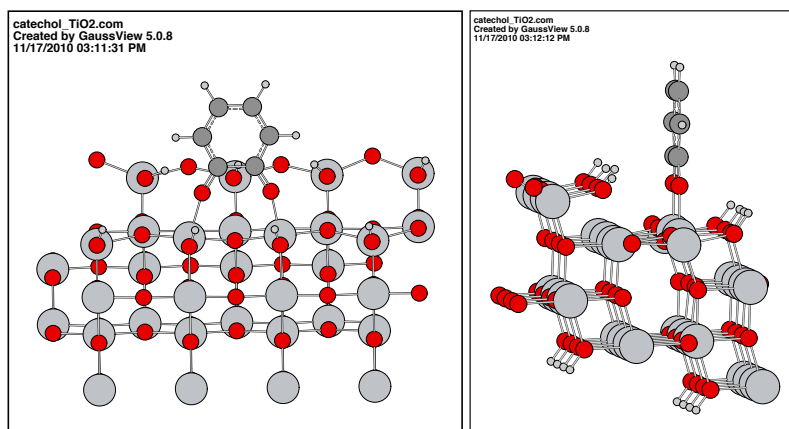


Figure 6: Nanostructure of  $\text{TiO}_2$ -anatase sensitized by catechol.

1. Open the structure file '`/dynamics-0.4/examples/TiO2_base.com`' for the  $\text{TiO}_2$  anatase lattice.
2. Click the 'Custom Fragment' icon (the 2nd one in the 1st row of the 'Builder' window), and then click the square button in the 3rd row. A new window 'Custom Fragments' pops out. Move cursor to the first line of the text part and click it. The first line is highlighted in blue, and then press the keyboard for 'Ctrl-N'. In this way, the active molecule (here,  $\text{TiO}_2$ ) set to be a custom fragment 'Fragment A'. Click 'Ok' to close the window of 'Custom Fragments'.
3. Click 'File' then 'Open' to open the file 'catechol.log',
4. Click the 'Custom Fragment' icon (the 2nd one in the 1st row of the 'Builder' window), you will see that 'Fragment A' is displayed on the button in the 3rd row, and the structure of  $\text{TiO}_2$  is visualized in the main window. If the structured shown in the main window is other existed custom fragment, you can click the button in the 3rd row to select the custom fragment of  $\text{TiO}_2$ .
5. In the main window, click the atom ('O') on  $\text{TiO}_2$  that you will use as the connecting point to the catechol molecule inside the window of '`G2:M1:V1 catechol.com`', and connecting atom is then highlighted in light blue color.
6. Once you highlighted the connecting point (the 'O' atom) of the  $\text{TiO}_2$  fragment, click on one of the O-atoms inside the window of '`G2:M1:V1 catechol.com`'. The catechol molecule is now attached to the  $\text{TiO}_2$  surface.
7. Adjust the catechol molecule to a proper position (by changing angles and dihedral angles) like the structure shown below, delete the extra 'O' atom on the  $\text{TiO}_2$  surface, and connect the other 'O' atom of catechol to the Ti atom of the  $\text{TiO}_2$  surface. Save the properly adjusted geometry to a file named '`catechol_TiO2.com`'.

A python script is available in the 'dynamics' directory to facilitate the conversion of the .com file to the crystallographic format required in the Yaehmop input file. The python script is called '`com_to_crystbind.py`' and when executed from the command line the name of the .com file should be included as an argument. For example

```
$ ~/dynamics-0.4/dynamics/com_to_crystbind.py catechol_TiO2.com
```

will generate the file '`catechol_TiO2.bind`' with the molecular structure in the required format for Yaehmop. The input for the 'Lattice' and 'Crystal Spec' keywords will be included in the generated file. The rest of the keywords needed for your desired calculation will need to be included using your favorite text-editor.

## References

- [1] Luis G.C. Rego and Victor S. Batista. *J. Am. Chem. Soc.*, 125:7989–7997, 2003.

- [2] Sabas G. Abuabara, Luis G.C. Rego, and Victor S. Batista. *J. Am. Chem. Soc.*, 127:18234–18242, 2005.
- [3] Sabas G. Abuabara, Clyde W. Cady, Jason B. Baxter, Charles A. Schmuttenmaer, Robert H. Crabtree, Gary W. Brudvig, and Victor S. Batista. *J. Phys. Chem. C*, 111:11982–11990, 2007.
- [4] William R. McNamara, Robert C. Snoeberger III, Gonghu Li, James M. Schleicher, Clyde W. Cady, Macarena Poyatos, Charles A. Schmuttenmaer, Robert H. Crabtree, Gary W. Brudvig, and Victor S. Batista. *J. Am. Chem. Soc.*, 130:14329–14338, 2008.